

Kalman Filter for Noise Reducer on Sensor Readings

Alfian Ma'arif^{a,1,*}, Iswanto^{b,2}, Aninditya Anggari Nuryono^{b,3}, Rio Ikhsan Alfian^d

^{a,d} Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

^b Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia

^c Department of Information Technology and Electrical Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia

¹ alfianmaarif@ee.uad.ac.id*; ² iswanto_te@umy.ac.id; ³ anindityanuryono@gmail.com; ⁴ rio1600022036@webmail.uad.ac.id

* corresponding author



ARTICLE INFO

Article history

Received May 23, 2019

Revised June 8, 2019

Accepted July 19, 2019

Keywords

Kalman Filter

Noise Reducer

Noise Filter

Sensor Data

ABSTRACT

Most systems nowadays require high-sensitivity sensors to increase its system performances. However, high-sensitivity sensors, i.e. accelerometer and gyro, are very vulnerable to noise when reading data from environment. Noise on data-readings can be fatal since the real measured-data contribute to the performance of a controller, or the augmented system in general. The paper will discuss about designing the required equation and the parameter of modified Standard Kalman Filter for filtering or reducing the noise, disturbance and extremely varying of sensor data. The Kalman Filter equation will be theoretically analyzed and designed based on its component of equation. Also, some values of measurement and variance constants will be simulated in MATLAB and then the filtered result will be analyzed to obtain the best suitable parameter value. Then, the design will be implemented in real-time on Arduino to reduce the noise of IMU (Inertial Measurements Unit) sensor reading. Based on the simulation and real-time implementation result, the proposed Kalman filter equation is able to filter signal with noises especially if there is any extreme variation of data without any information available of noise frequency that may happen to sensor-reading. The recommended ratio of constants in Kalman Filter is 100 with measurement constant should be greater than process variance constant.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

High-sensitivity sensors are very vulnerable to noise when reading data from environment [1]. Vibration, wind pressure, are some examples of external disturbances which cause noise on data-reading. Meanwhile, most systems nowadays require high-sensitivity sensors to increase its system performances [2] [3]. In example, high-sensitivity accelerometer and gyro are needed in robot systems [4] [5] [6], quad-rotors [7] and maglev systems [8]. These sensors are commonly being uncovered from external disturbances such as wind pressure and vibration of rotors. This can result noise on data-readings. For example, Fig. 1 shows the unfiltered result of ADC readings of an accelerometer of an unmoving balancing wheel. Based on Fig. 1, it can be known that sensor readings are affected by noise or disturbances since the result are varying for an unmoving robot.

Noise on data-readings can be fatal since the real measured-data are used as an input to a controller. High-sensitivity sensors contribute to the sensitivity of a controller. If noise is calculated as the measurement data, the control system cannot achieve its best performances or even fail to control the system. As for previous case of a varying data result taken from an accelerometer of unmoving balancing robot, the noise could affect the controller's performance as the controller will respond to the 'error', which causes by the difference of the result from sensor's readings with the referenced

acceleration. The controller may become too aggressive while controlling the robot's movement. Furthermore, this will result to bad system's performances i.e. uneven trajectory of the robot's movement.

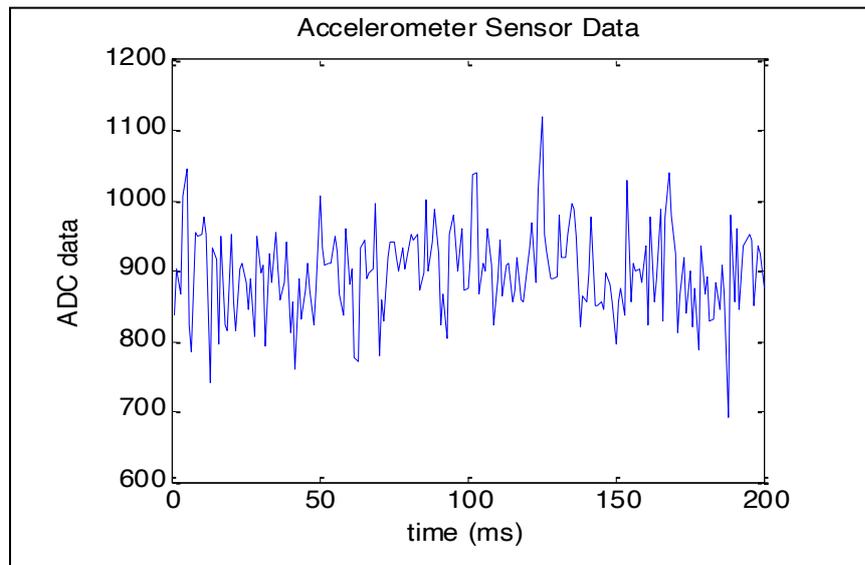


Fig. 1. Unfiltered accelerometer result of unmoving robot

Ideally, all of the noise from the readings should be cancelled for high-sensitivity sensors. However, it is practically impossible to cancel all the noise since it happens due to external environment natures. Other options available are to reduce the noise or to separate the noise from the real measurement data.

Most common filters, such as Low Pass Filter (LPF), High Pass Filter (HPF) and Band Pass Filter (BPF) are using frequency to differentiate the noise and the real-measurement data. This is the simplest method but has some disadvantages. First, we need to know the frequency of the noise. This is applicable when the noise has distinctive frequency with the real-measurement data. However, this is not applicable when the noise happens in all range of frequencies.

Kalman filter was proposed by R. E. Kalman in 1960 [9], and also called Linear Least Mean Squares Estimator (LLSME) or Linear Quadratic Estimation (LQE) [10]. It has been applied to many fields such as in robotics [11] [12], control system [13] [14], battery management system (BMS) [15] [16], dynamic process control [17], flood prediction [18], wind direction prediction [19], tracking objects [20], navigation systems [7], and computer vision application (stabilizing depth measurements, fusing data).

Kalman filter is popular for having easy computation, memory requirements and good capability on overcoming noises. It is state technique estimation that can extract information from noisy data [21]. Hence, Kalman Filter is best to use for general noise reducer on sensor-reading, especially when the information of noise's frequency that may happen to the sensor-reading is unavailable.

Despite the advantages, it is challenging to design the required equation of Kalman Filter's design for noise reducer. Moreover, it is also important to be careful while determining some parameters such as measurement constant and process variance constant because it affects the filter result much. The paper will discuss on how to specifically design the Kalman Filter equation and parameters for noise reducer on sensor readings. The Kalman Filter equation will be theoretically analyzed and designed based on its component of equation. Also, some values of measurement and variance constant will be simulated and then the filtered result will be analyzed to obtain best suitable values of the parameters.

The paper will be arranged into four sections as follows. The first section will be an introduction. The second part will discuss the Kalman Filter and the design process on purpose to reduce signal's noise. The third section is result and discussion about Matlab and Arduino testing. Then last section will provide the conclusions.

2. Method

2.1. Kalman Filter

Kalman filter is an algorithm which is used to predict (or to estimate) the next result based on the previous data. It is not categorized as common filters such as LPF, HPF, and BPF. It is basically an estimator to predict any state or part in the signal which contains signal. The result of the estimation process is similar with eliminating noise from the signal, which is why it is called as Kalman Filter.

Systems in Kalman Filter are assumed as linear systems. Kalman Filter minimalizes the average estimation error square for stochastic linear systems using linear noises sensor. It also minimalizes estimation error square function for linear dynamic systems by white measurement and noise disturbance. It functions as to estimate states on dynamic systems and also to analyze system's performance [10].

There are various types of Kalman Filter, such as standard Kalman Filter [4], Extended Kalman Filter [22], Unscented Kalman Filter [23] and Ensemble Kalman Filter [24]. Standard Kalman Filter is the simplest while the other types are modified for more complicated tasks. The paper will use standard Kalman filter since it contains enough part of equation for noise reducing.

Kalman Filter has two parts, the predict part and the update part. The standard Kalman Filter equation is shown in (1) – (5).

Predict:

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t \quad (1)$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \quad (2)$$

Update:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - H_t \hat{x}_{t|t-1}) \quad (3)$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \quad (4)$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} \quad (5)$$

where x is estimated state, F is state transition matrix, u is control variables, B is control matrix, P is state variance matrix, Q is process variance matrix, y is measurement variables, H is measurement matrix, K is Kalman gain, R is measurement matrix, $t|t$ is current time period, $t-1|t-1$ is previous time period, and $t|t-1$ is intermediate steps.

Equations (1) to (5) can be called as Kalman Filter system model and its purpose is not defined yet. Therefore, they can be modified based on the designed purpose and how complex the system will be. To implement Kalman Filter algorithm, so that it can be used to reduce noise of sensor-readings, some adjustments for the conditions are needed. Those adjustments are as follows.

1. Predicting the state

On this stage, adjustments are done in (1) by giving the score $F_t = 1$ because there is no state transition. Thus, reducing the system's input component B_t because the used system does not have any input u_t . The adjusted equation is shown in (6).

$$x_{t|t-1} = x_{t-1|t-1} \quad (6)$$

2. Predicting the error

Since $F_t = 1$, then (2) becomes (7).

$$P_{t-1} = P_{t-1|t-1} + Q_t \quad (7)$$

3. Updating the state value

From (3), $H_t = 1$ since the sensor data that will be filtered is only consisted of one sensor reading. Hence, the equation can be written as (8).

$$x_{t|t} = x_{t|t-1} + K_t(y_t - x_{t|t-1}) \quad (8)$$

4. Calculating the gain of Kalman

Since $H_t = 1$, then (4) can be written as (9).

$$K_t = P_{t|t-1}(P_{t|t-1} + R)^{-1} \quad (9)$$

5. Updating the error value

Since $H_t = 1$, then (5) can be turned into (10).

$$P_{t|t} = (1 - K_t)P_{t|t-1} \quad (10)$$

2.2. Kalman Filter for Noise Reducer

After the adjustments are done, Kalman Filter equation for reducing noise of sensor-reading can be rewritten as (11) – (15).

Predict:

$$x_{t|t-1} = x_{t-1|t-1} \quad (11)$$

$$P_{t-1} = P_{t-1|t-1} + Q_t \quad (12)$$

Update:

$$x_{t|t} = x_{t|t-1} + K_t(y_t - x_{t|t-1}) \quad (13)$$

$$K_t = P_{t|t-1}(P_{t|t-1} + R)^{-1} \quad (14)$$

$$P_{t|t} = (1 - K_t)P_{t|t-1} \quad (15)$$

2.3. Algorithm

The algorithm of Kalman filter, which is based on (11) – (15), is illustrated as flowchart as in Fig. 2. It can be implemented through Matlab, Arduino or another programming sources. After the program starts, some initialization must be made for some variables and constants. In this step, the process variance constant Q and the measurement constant R must be chosen properly. Initially, the values of the estimated state variable x_t can be set to zero and the state variance variable P_t can be set to one. The next step is to make the sensor data as the input of the Kalman filter as unprocessed signal. The third step is to calculate the predict part which consists of the prediction estimated state variable $x_{t_predict}$ and the prediction state variance variable $P_{t_predict}$.

The fourth step is to calculate the update part, which actually consists of the Kalman gain K_t the updated estimated state variable x_t , and the updated state variance variable P_t . In the update part, the filtered data is the updated estimate state variable x_t and will be shown as the output of the algorithm. Some other values such as the last estimated state and the last state variance is also saved as previous data for next iteration as the looping process is doing lifetime until the system is terminated. The sample source is available in the appendix.

3. Results and Discussion**3.1. Matlab Implementation**

The research is done by doing simulation in Matlab. The research scenario is by making a signal with noise then it will be filtered by using Kalman Filter. Both data, before and after it gets filtered,

are compared to find out the Kalman Filter performances. Hence, some parameters in Kalman Filter, which are R and Q matrices, is changed by some values and then is analyzed to seek which ones are the optimal ones to get the best result of the filter.

The test results are shown in Fig. 3, Fig. 4 until Fig. 8. The X-axis shows time and Y-axis shows the data. Green-colored signal is input signal which has original data, some noises, disturbance and extreme variation value. The red-colored signal is the input signal result after getting filtered by Kalman Filter. In testing, it is assumed that the data from random signal is the values of sensor reading with big noise and disturbance.

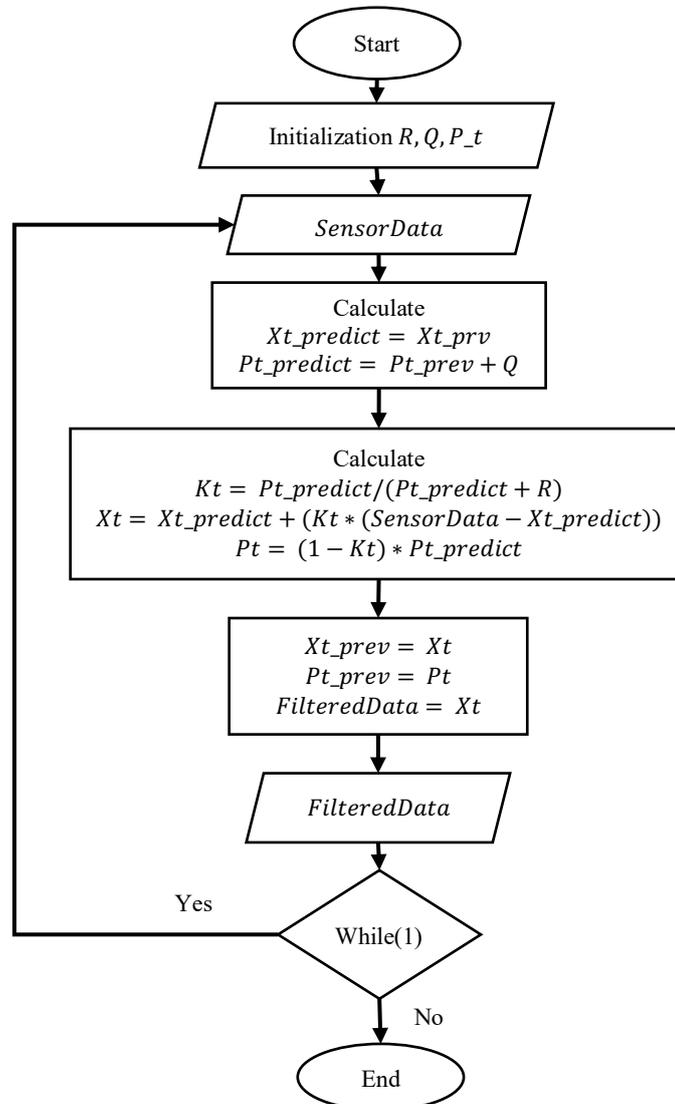


Fig. 2. Kalman Filter Flow Chart

The first examination is in Fig. 4 where using the measurement constant, $R = 1$ and the variance constant, $Q = 1$. Based on Fig. 4 the input signal only gets filtered a bit when it is compared to the input signal. Moreover, the values are varying drastically. This may result as very aggressive system response.

The second examination is shown in the Fig. 5 where the measurement constant $R = 1$ and the variance constant $Q = 0.1$. Based on Fig. 5, the input signal starts to get filtered better than before. However, the values are still varying drastically although not as big as previous test filtered result. This may still result in a very aggressive system response. Parameter values from both of this and previous examination are not suitable to be implemented into real systems.

The third and fourth examination respectively are shown in the Fig. 6 using $R = 1$ and $Q = 0.01$ and Fig. 7 using $R = 10$ and $Q = 0.1$). The both examination is the best two of filter results because the noise is able to be reduced yet the original data characteristics are still preserved. Similar results are shown by those figures, most possibly because of the same ratio difference of R and Q values. Based on the examination, both of the parameter values are suitable to be implemented into the real system.

The fifth examination is shown in Fig. 8 using $R = 100$ and $Q = 0.1$. Based on Fig. 8, the input signal loses its original data so that it can be said that the ratio difference of R and Q should not be greater than 1000 or it will give overly-filtered result. Thus, the parameter is not suitable to be implemented into real system because of the loss of original data characteristics.

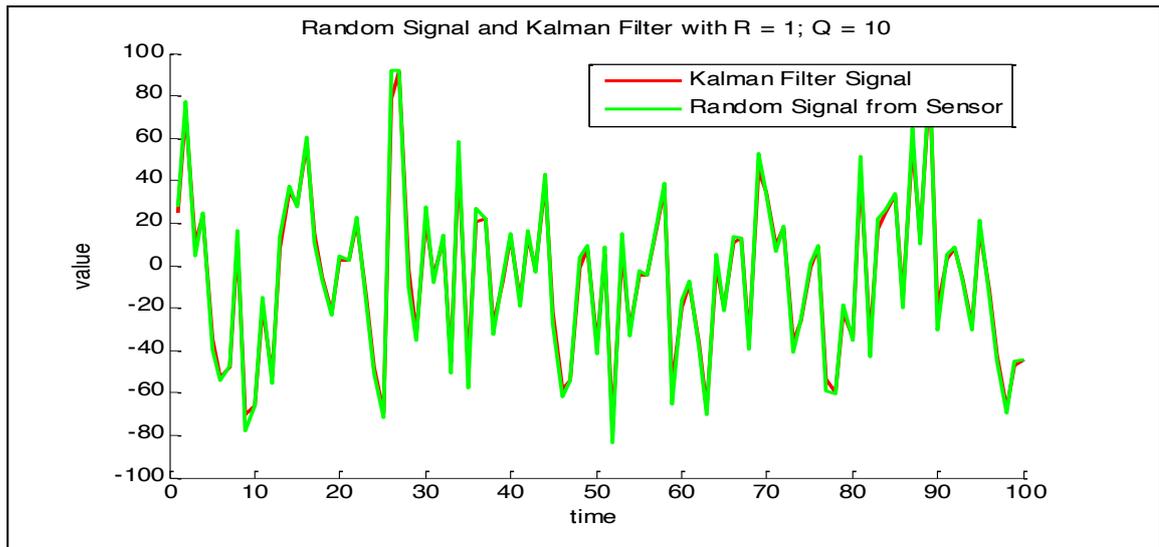


Fig. 3. First result with $R = 1$ and $Q = 10$

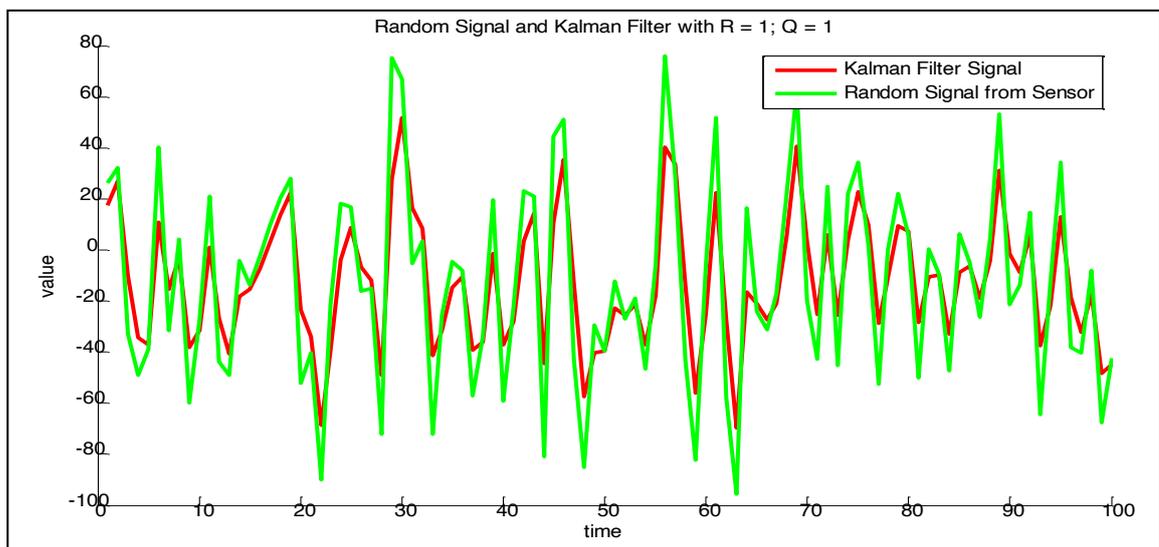
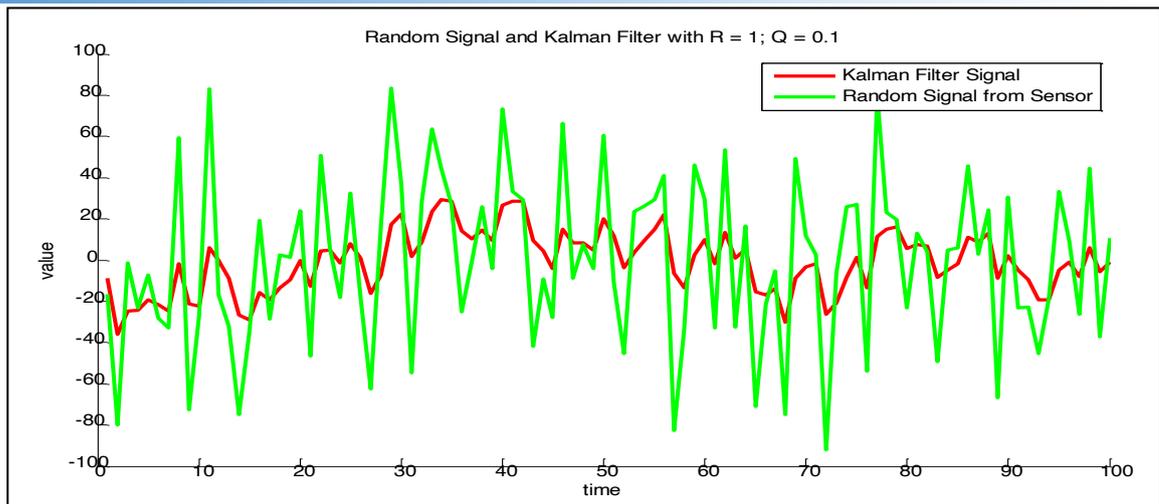
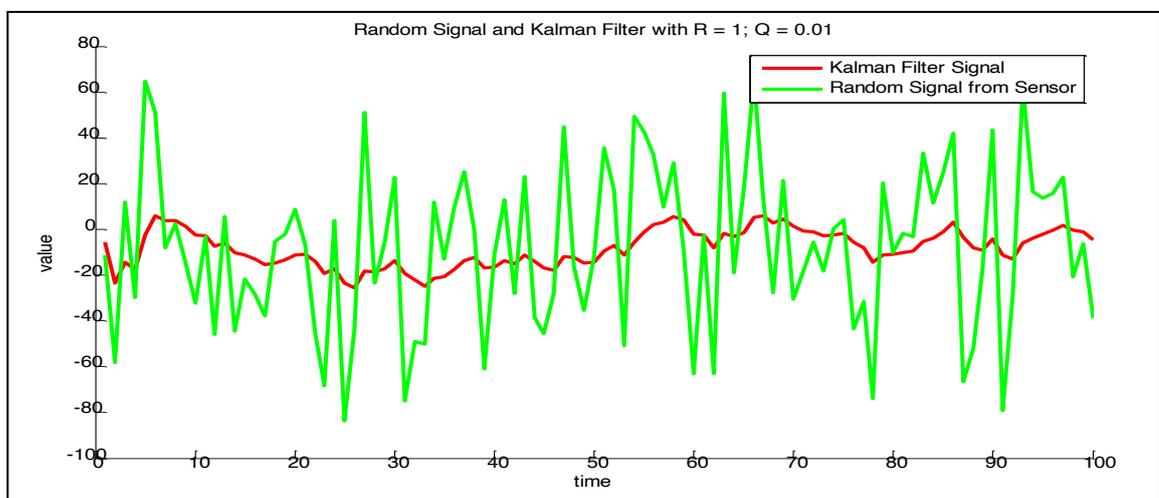
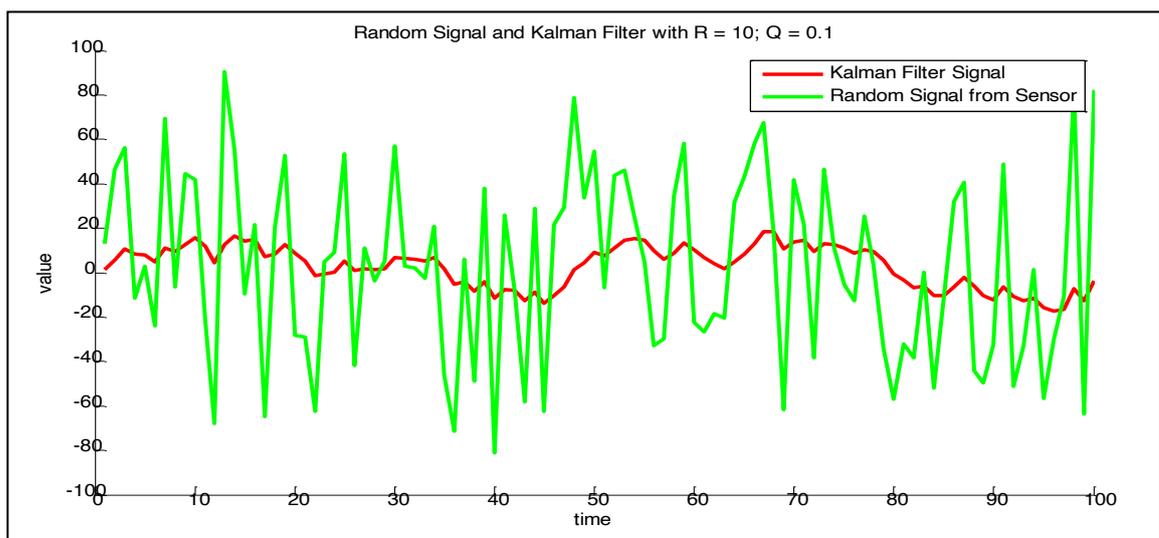


Fig. 4. Second result with $R = 1$ and $Q = 1$

**Fig. 5.** Third result with $R = 1$ and $Q = 0.1$ **Fig. 6.** Fourth result with $R = 1$ and $Q = 0.01$ **Fig. 7.** Fifth result with $R = 10$ and $Q = 0.1$

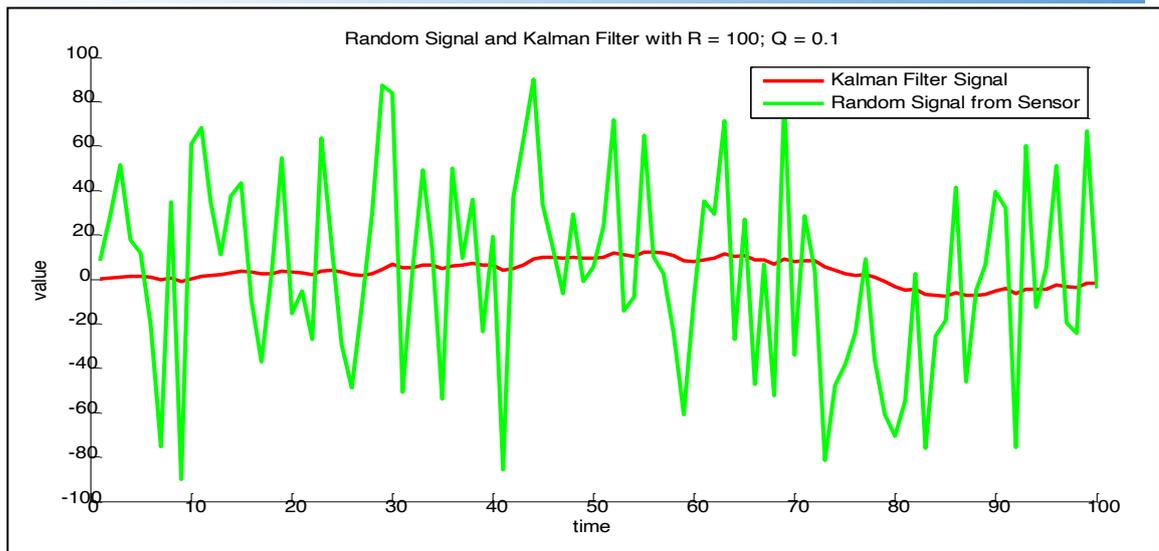


Fig. 8. Sixth result with $R = 100$ and $Q = 0.1$

Based on the examination from Fig. 3 until Fig. 7, those constants highly affect the noise reducer. The bigger the ratio between R and Q , the bigger the noise damping effect. To investigate the damping effect, the filter result is then compared to the real values. The difference then can be seen as an error equation which can be written as follows

$$\bar{e} = |x_n - y_n| \quad (16)$$

The equation \bar{e} is known as mean error where x is the data values and y is filter results. The calculations from the equation are shown as in Table 1 on mean error column. Based on Table 1, it can be seen that the bigger difference between R and Q make bigger mean error values. Besides, the same R and Q difference values result in similar value of mean error, whatever the values of R and Q . By comparing Table 1 with previous analysis from the figures, best parameters which provide result with its original data characteristics have mean error values range from 40 to 55.

Table 1. Mean Error and Ratio Value

Experiment Number	Kalman Filter Parameter Value		R and Q Ratio	Mean Error
	R	Q		
1	1	1	1	26.0677
2	1	0.1	10	44.7392
3	1	0.01	100	53.4466
4	10	0.1	100	53.4541
5	100	0.1	1000	56.9959

3.2. Arduino Implementation

After simulation using Matlab, the next step is to implement Kalman Filter in Arduino to reduce the noise of IMU (Inertial Measurements Unit) sensor. The IMU sensor consists of Accelerometer Sensor and Gyro Sensor. The results of accelerometer sensor noise reducer are shown by Fig. 9 and Fig. 10. The X-axis on those figures is the n -th data and Y-axis is the 10-bit-ADC sensor data. Green-colored line represents the sensor data and red-colored line represents the result of Kalman filter. Parameters values of Fig. 9 are $R=10$ and $Q=0.1$. Meanwhile, $R=10$ and $Q=0.01$ are used on Fig. 10.

Fig. 9 and Fig. 10 shows that the noise is able to be reduced by using Kalman Filter. The ratio of R and Q on Fig. 9 is 100 and 1000 on Fig. 10. The result from Fig. 9 provides better performance in preserving the original data characteristics than the result from Fig. 10. The bigger the ratio of R and Q make smoother filter result. However, if the ratio difference is too wide, the original data loss may

happen as in Fig. 10. The loss of original data characteristics should be avoided to present real measurement of the sensor.

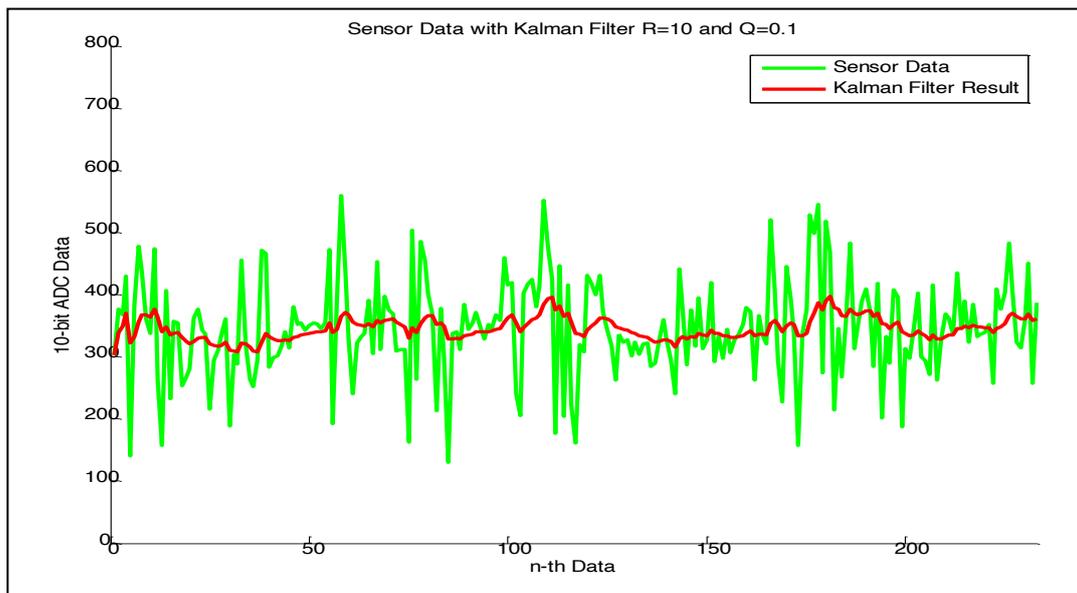


Fig. 9. Seventh result with $R = 10$ and $Q = 0.1$

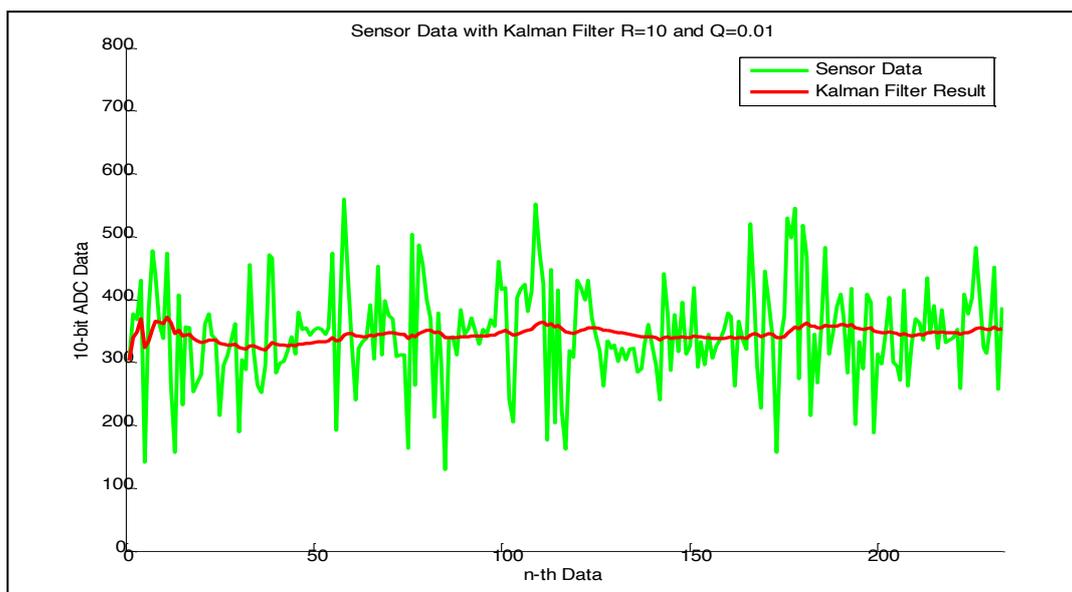


Fig. 10. Eighth result with $R = 100$ and $Q = 0.1$

4. Conclusion

Based on the simulation and test result, a proposed equation of modified Standard Kalman Filter is able to reduce noise in sensor readings. The extreme increase and decrease of the result can be damped so that it will not happen on the output. The damping value of the filter depends on the constants difference. If R and Q has big ratio difference like 1000, 10000, and so on, the damping will get bigger and can reduce the original data. Meanwhile, if the ratio difference of R and Q has smaller difference like 10, 1, 0.1 and so on, the damping will get smaller till it does not have any damping effect. The recommended ratio difference of R and Q values is 100.

Acknowledgement

The authors would like to express sincere gratitude to the editor and blind reviewer for their advice and suggestion to improve the quality of the paper.

Appendix

a. Matlab Code Kalman Filter Sample Program

```

clear; close all; clc;

xk = 0; Pk = 0; R = 100; Q = 1;

for kk=1:100
    zk = (rand-rand)*100;
    Pk = Pk + Q;
    Kk = Pk / ( Pk + R );
    xk = xk + (Kk * (zk - xk) );
    Pk = (1-Kk)*Pk;
    buff_xk(kk) = xk;
    buff_zk(kk) = zk;
end

figure(1)
hold on
plot(buff_xk,'r' , 'linewidth',2 )
plot(buff_zk,'g' , 'linewidth',2 )
title('Random Signal and Kalman Filter with R = 100; Q = 1')
xlabel('time'); ylabel('value')
legend('Kalman Filter Signal','Random Signal from Sensor')

```

b. Arduino Code Kalman Filter Program

```

float SensorData, KalmanFilterData;
float Xt, Xt_update, Xt_prev;
float Pt, Pt_update, Pt_prev;
float Kt, R, Q;

void setup() {
    Serial.begin(9600);
    R=100; Q=1; Pt_prev=1;
}

void loop() {
    SensorData = analogRead(A0);
    Xt_update = Xt_prev;
    Pt_update = Pt_prev + Q;

    Kt = Pt_update / (Pt_update + R);
    Xt = Xt_update + ( Kt * (SensorData - Xt_update));
    Pt = (1 - Kt) * Pt_update;

    Xt_prev = Xt;
    Pt_prev = Pt;

    KalmanFilterData=Xt;

    Serial.print (SensorData,3);
    Serial.print (",");
    Serial.print (KalmanFilterData,3);
    Serial.println();

    delayMicroseconds (100);
}

```

References

- [1] S. A. Quadri and O. Sidek, "Error and Noise Analysis in an IMU using Kalman Filter," *International Journal of Hybrid Information Technology*, vol. 7, no. 3, pp. 39–48, 2014.
- [2] G. Pang and H. Liu, "Evaluation of a low-cost MEMS accelerometer for distance measurement," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 30, no. 3, pp. 249–265, Mar. 2001.
- [3] M. Safitri, A. Cahyadi, E. Firmansyah, and J. Grafika, "Estimasi Posisi UAV dengan Kalman Filter," no. September, pp. 240–245, 2015.
- [4] N. G. M. Thao, D. H. Nghia, and N. H. Phuc, "A PID backstepping controller for two-wheeled self-balancing robot," in *2010 International Forum on Strategic Technology, IFOST 2010*, 2010, pp. 76–81.
- [5] M. Kamaludin and W. S. Aji, "Manuver Robot Manual Menggunakan PID pada Robot Manual KRAI 2018," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 3, p. 91, 2019.
- [6] K. Kunal, A. Z. Arfianto, J. E. Poetro, F. Waseel, and R. A. Atmoko, "Accelerometer Implementation as Feedback on 5 Degree of Freedom Arm Robot," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, pp. 31–34, 2020.
- [7] Lasmadi, A. Cahyadi, S. Herdjunto, and R. Hidayat, "Inertial navigation for quadrotor using kalman filter with drift compensation," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 5, pp. 2596–2604, Oct. 2017.
- [8] A. Ma'arif, A. imam Cahyadi, and O. Wahyunggoro, "CDM Based Servo State Feedback Controller with Feedback Linearization for Magnetic Levitation Ball System," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 3, p. 930, Jun. 2018.
- [9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [10] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. Wiley, 2015.
- [11] L. Lasmadi, "Attitude Estimation for Quadrotor Based on IMU with Kalman-Filter," *Conference SENATIK STT Adisutjipto Yogyakarta*, vol. 4, no. 0, pp. 351–358, Nov. 2018.
- [12] I. Maulana and N. S. Widodo, "Sistem Pengolah Musik Sebagai Kontrol Gerak Robot Humanoid," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 2, p. 46, 2019.
- [13] A. Ma'arif, A. I. Cahyadi, O. Wahyunggoro, and Herianto, "Servo state feedback based on Coefficient Diagram Method in magnetic levitation system with feedback linearization," in *2017 3rd International Conference on Science and Technology - Computer (ICST)*, 2017, pp. 22–27.
- [14] A. Ma'arif, H. Nabila, Iswanto, and O. Wahyunggoro, "Application of Intelligent Search Algorithms in Proportional-Integral-Derivative Control of Direct-Current Motor System," in *Journal of Physics: Conference Series*, 2019, vol. 1373, no. 1.
- [15] P. A. Topan, M. N. Ramadan, G. Fathoni, A. I. Cahyadi, and O. Wahyunggoro, "State of Charge (SOC) and State of Health (SOH) estimation on lithium polymer battery via Kalman filter," in *Proceedings - 2016 2nd International Conference on Science and Technology-Computer, ICST 2016*, 2017, pp. 93–96.
- [16] G. Fathoni, S. A. Widayat, P. A. Topan, A. Jalil, A. I. Cahyadi, and O. Wahyunggoro, "Comparison of State-of-Charge (SOC) estimation performance based on three popular methods: Coulomb counting, open circuit voltage, and Kalman filter," in *Proceedings of the 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology, ICACOMIT 2017*, 2017, vol. 2018-January, pp. 70–74.
- [17] M. Shyamalgowri and R. Rajeswari, "Unscented Kalman filter based nonlinear state estimation case study-Nonlinear process control reactor (Continuous stirred tank reactor)," in *Proceedings of the 10th International Conference on Intelligent Systems and Control, ISCO 2016*, 2016.
- [18] F. A. Ruslan, A. M. Samad, and R. Adnan, "Modelling of flood prediction system using hybrid NNARX and Extended Kalman Filter," in *Proceedings - 2017 IEEE 13th International Colloquium on Signal Processing and its Applications, CSPA 2017*, 2017, pp. 149–152.
- [19] Z. Dzulfikri, N. Nuryanti, and Y. Erdani, "Design and Implementation of Artificial Neural Networks to Predict Wind Directions on Controlling Yaw of Wind Turbine Prototype," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, 2020.
- [20] Y. Wu, Y. Sui, and G. Wang, "Vision-Based Real-Time Aerial Object Localization and Tracking for UAV Sensing System," *IEEE Access*, vol. 5, pp. 23969–23978, Oct. 2017.
- [21] Y. Pei, S. Biswas, D. S. Fussell, and K. Pingali, "An elementary introduction to Kalman filtering," *Communications of the ACM*, vol. 62, no. 11, pp. 122–133, Nov. 2019.

- [22] S. Nurmaini and S. Pangidoan, "Localization of Leader-Follower Robot Using Extended Kalman Filter," *Computer Engineering and Applications Journal*, vol. 7, no. 2, pp. 95–108, Jul. 2018.
- [23] H. Khazraj, F. Faria Da Silva, and C. L. Bak, "A performance comparison between extended Kalman Filter and unscented Kalman Filter in power system dynamic state estimation," in *Proceedings - 2016 51st International Universities Power Engineering Conference, UPEC 2016*, 2016, vol. 2017-January, pp. 1–6.
- [24] S. Afrasiabi, M. Afrasiabi, M. Rastegar, M. Mohammadi, B. Parang, and F. Ferdowsi, "Ensemble kalman filter based dynamic state estimation of PMSG-based wind turbine," in *2019 IEEE Texas Power and Energy Conference, TPEC 2019*, 2019.